Kubernetes

**What are Kubernetes?**

Open – source container orchrstration tool, developed by Google.

Basically K8’s helps in manage applications that are made up of hundereds of containers & helps to manage them in different env.

**What is the need of container orchestration technologies?**

container orchestration technologies are crucial for managing large-scale containerized environments. They provide a range of benefits and capabilities that make it easier to deploy, manage, and scale applications effectively

**what features does orcehestation tool offers?**

* High Availability ,or down time
* Scaleability or high performance
* Desaster recovery – backup and restore

**1.1 Kubernet Comopents**

1. **Node:** Node is a simple server or virtual machine

**(**In Kubernetes (often abbreviated as K8s), a "Node" is a worker machine in the Kubernetes cluster. It can be a physical machine or a virtual machine, depending on the cluster setup. Nodes are responsible for running containers, which are grouped into Pods.)

1. **Pod** : basic components and smallest unit of k8,s is pod

* Pod is abstraction over container,pods creates this running env or layer on top of this container.(reson is k8,s Abstract away the container technologies so that we can replace them
* **How this pods inside server comunitacets**: K8,s provide unique IP address to each pods and each pod can communicate with each other usig this IP address.

*(A "Pod" is the smallest deployable unit in Kubernetes. It represents a single instance of a running process in your cluster. A Pod can contain one or more containers that are tightly coupled and share resources, such as network and storage, and can communicate with each other using localhost. Pods are created and managed by the Kubernetes API server)*

* If pod inside node dies b/c of connection loss then new pod will be created with new ip address on re-creation, and obvisuly inconvenient to adjust ip address on every pod restart. This can be over come using other k8,s components called **Services. & Ingres**

pod

My-app

ip

Container

DB

ip

Node1

Node 1

Node1

1. **Services:** Services are static or permanent IP address that are attached to each pods, so each pods inside node will be having its own Services.

* Life cycle of pods and services are not connected. Even if pod dies services will stay so no need to change the endpoint.
* ***A Service is an internal abstraction used to expose a set of Pods as a network service. It provides a stable endpoint to access a group of Pods, usually by using a label selector to identify the Pods it targets.***

**Node1**

MY-app

pod

Services

cont

DB

services

1. **Ingress :**

* Ingress is an API object used to manage external access to services in a Kubernetes cluster. It provides HTTP and HTTPS routing to services based on the request's host, path, or other criteria.

1. **ConfigMap:** when there is slight changes in pods ex: like url or other in that case we need to re-build new –version of the application , push to repository and again we need to pull new image to pod and restart the whole thing. To overcome this we have K8,s **ConfigMap.**

* ConfigMap contains external configuration data of our application like URL,or some other services that we use .

1. ***A ConfigMap is an API object used to store non-sensitive configuration data in key-value pairs. It provides a way to inject configuration data into Pods at runtime.***
2. ***ConfigMaps are typically used to store configuration files, command-line arguments, environment variables, or any other configuration data that doesn't contain sensitive information.***
3. ***Configuration data stored in ConfigMaps can be consumed by Pods either as environment variables or as mounted volumes, allowing applications to access the configuration data easily***
4. **Secret:**

A Secret is an API object used to store sensitive data, such as passwords, API keys, and certificates, in an encrypted format. Secrets are stored in etcd, the key-value store used by Kubernetes, and are only accessible to the nodes and components that need them.

1.  ***Secrets provide a way to securely store and distribute sensitive information to Pods without exposing it in plain text.***
2. *** Similar to ConfigMaps, Secrets can be mounted into Pods as files or exposed as environment variables. However, when a Secret is mounted as a file, it is stored in a tmpfs volume, which is a memory-backed filesystem, providing additional security by not persisting the sensitive data to disk***.
3. **Volumes :** Volumes provide a way for containers within the same Pod to share data or persist data beyond the lifetime of the containers themselves. When we re-start our container or pod we will be having all the data stored somewhere in local or in cloud done through volumes.

**Kubernet cluster dono’t manage the storing of data. We Kubernet user should take responsibility to backing up the data and mnageing it make sure it is safe in hardware**

1. **Deployment :** When there is application crash or application down time user are facing with single node and pod , this can be over come by replicating the Node for multiple instances where pod inside the each Node will share comman services.

**Service are having 2 functinalities:**

* It carries permanent ip address
* Act as load balancer – which means service will catch the request and forward it to which ever pod is busy.
* In order to create replica of pod we no need to creat pod instead we need to define blueprint for application pod & specify how replica of application pod we need to run. And that component or blueprint is called as Deployement.
* We can scale up or sacle down number of replicas of pods .
* Pods are Abstraction on top of containers and Deployemets are Abstraction on top of pods which make easy to communicate and replicate them.

1. **Statefulset : we can’t replicate database b/c if we do that there will be shared db and we need to manage which pods are reading from db which are writteing to db or currently in us which leads to inconsistent data**

* To replicate applications like DATABASE we use kuberenetes components called Statefulset
* MYsql, mongoDb , and other Db should be created using statefulset.

**Difference between Deployment and Statefulset**

* **Deployment:** Primarily used for stateless applications, where the Pods do not need to maintain any state between restarts or rescheduling. Examples include web servers, APIs, and microservices.
* **StatefulSet:** Used for stateful applications, where the Pods require persistent storage and stable network identities. Examples include databases, distributed systems, and other applications requiring stable storage and network identities.

Note :

Nodes run Pods, and each Node can run multiple Pods. Kubernetes manages the scheduling and placement of Pods onto Nodes based on resource requirements, affinity/anti-affinity rules, and other constraints specified in the Pod's configuration.

**1.2 Kubernetes Architecture**

* **Worker node**
* **3 important components in Worker nodes**
* **Kubelet**
* **Kube – proxy**
* **Container – runtime**

1. **How to interact with this cluster**

* **How to decide on which pod or db pod should be scheduled ?**
* **If replica dies what process would monitors it ?**
* **And then reschedule /re-start pod again?**
* **Or when we add new server how does it join the cluster to become an other node and get other pods and components created on it ?**

**Ans : all this managing process are done by master node**

**4 processes runs on every master node**

**1 Api server :** API server act as an entry point to cluster, if we want to deploye any application into cluster 1st API server will get inital request about updates into cluster and act as gatekeeper it make sure only authenticated and authorized request get through cluster.

* How it API server works ?

If we want to create any services, or schedule new pods, or deploye application 1st we need to communicate with API server , API server will validate and make sure everything is ok it will forward request to other comopents and in order to sechudle pod or create component that we requested.

**2 Scheduler :** scheduler just decides on which node new pod should schedule this request are collected by kublet and starts the execution on that node

1. **Controller manager** : it will detect cluster state change(pod dies, crashes) and again resecudle the process

**4 etcd :** act as cluster brain, cluster changes gets stored in the key-value pair

**1.3 Minikube & Kubectl – local set-up :**

**Production server** – multipal master & worker node.

**MiniKube :** contains one master processes and one worker processes inside single node where docker container runtime is pre-installed **.**

Minikube will create virual box on laptop and this node will run on the virtual box.

**Kubctl :** it is a command tool used to intract with virtual node to create pods & other kubernetes comopents on node .

**Installing Kubctl :**

[**https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/**](https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/) **official website**

**Select operating system (windos):**

### Install kubectl binary with curl on Windows

1. Download the latest 1.30 patch release: [kubectl 1.30.0](https://dl.k8s.io/release/v1.30.0/bin/windows/amd64/kubectl.exe)

Click on above kubectl 1.30.0 link

3 Copy and cut the downloaded file and paste it to desired location

4 set environment variaple ---- go to path ----copy the path of exe file where you have pasted –click on okm

5 to check installed peoerly or not open cmd – type kubectl – need to get info